

Chapter 2

Data Assimilation

The goal of this Chapter is to develop a comprehensive set of estimation equations for assimilating remote sensing data into a model of land surface dynamics. The hydrologic model itself is described in detail in Chapters 3 and 4.

In Section 2.1 we first define the state and measurement equations in a general form. From this general formulation, we then derive the nonlinear estimation equations (Euler-Lagrange equations) using a variational technique. Sections 2.2 and 2.3 provide an overview of the representer algorithm which is used to solve for the estimates.

The focus of Section 2.4 is to derive the posterior covariances of the state and the measurements within the representer approach. In Section 2.5 we briefly discuss the nature of the representer approach as a data space search engine and the opportunities for data compression and for the a posteriori assessment of the observing system.

2.1 General Formulation of the Estimation Problem

2.1.1 State and Measurement Equations

We formulate the state in vector form, assuming that the model equations have been discretized in space but not in time. In other words, the components of the state vector correspond to state variables at discrete spatial nodes but depend continuously on time. There are two parts to the state vector: the vector $X(t)$ of length N_X which obeys a set of implicit algebraic equations, and the vector $Y(t)$ of length N_Y which obeys a set of ordinary differential equations. In a land surface model, the storage terms of some canopy states, for example the canopy temperature, are typically neglected. Therefore such canopy states obey diagnostic (algebraic) equations. Soil moisture and temperature, on the other hand, have significant memory through storage of water and energy in the soil. Such prognostic variables are subject to ordinary differential equations. For details on the exact definition of X and Y within the land surface model see Section 4.2. Together, X and Y obey the *state equation*

$$\begin{aligned} 0 &= \phi(X, Y; \alpha) + D_\nu P_\nu \nu \\ \frac{\partial Y}{\partial t} &= \varphi(X, Y; \alpha) + D_\omega(Y) P_\omega \omega \end{aligned} \tag{2.1}$$

The operators $\phi(X, Y; \alpha)$ and $\varphi(X, Y; \alpha)$ depend nonlinearly on the state and on the uncertain parameter vector α of length N_α . In the soil moisture application, these parameters

could for example be the saturated hydraulic conductivities. The state equation also includes the process noise terms $\nu(t)$ and $\omega(t)$, which are time-dependent vectors of length N_ν and N_ω , respectively.

The diagonal matrices D_ν and D_ω of size N_X and N_Y , respectively, account for scaling between the states and the process noise. To facilitate the estimation, it is important that all variables be scaled. On the other hand, we would like to directly relate the process noise to a physical flux. As a result, it is necessary to account for scaling between the states and the process noise. We also include a formal dependence $D_\omega(Y)$ of the scaling matrix on the state, which we will need for the soil moisture application. From the spatial discretization, we usually get a matrix multiplying the time derivative. We include this factor in the operator φ and the matrix D_ω .

Finally, the process noise does not necessarily affect all components of the state vector. The $N_X \times N_\nu$ matrix P_ν and the $N_Y \times N_\omega$ matrix P_ω serve the purpose of projecting the process noise onto just those components of the state vector that we consider subject to model errors. In this sense, the choice of P_ν and P_ω partially reflect our assumptions on the model's shortcomings. Both P_ν and P_ω contain ones and zeros only.

The state equation (2.1) is subject to the *initial condition*

$$Y|_{t=0} = Y_0(\beta) \quad (2.1a)$$

which is parameterized by the uncertain vector β of length N_β . Section 4.3 explains why we need such a nonlinear parameterization. Without loss of generality, we have set the initial time to zero. The final time of the estimation interval is denoted with t_f .

Measurements, and in particular remote sensing data, are not necessarily direct observations of the state. We therefore introduce a nonlinear *measurement equation*.

$$Z = M[X, Y] + v \quad (2.2)$$

All N_Z individual measurements are collected into the data vector Z and are corrupted by the measurement noise v . The measurement operator $M[\cdot]$ is a vector-valued, nonlinear functional operating on vector-valued functions (e.g. the state vector). For later use, we rewrite the measurement operator according to

$$M[X, Y] = \int_0^{t_f} [\delta] f(X, Y) dt \quad (2.3)$$

where

$$[\delta] \equiv \text{diag} \{ \delta(t - t_1), \delta(t - t_2), \dots, \delta(t - t_{N_Z}) \} \quad (2.3a)$$

Both $M[X, Y]$ and $f(X, Y)$ are vectors of length N_Z . Whereas $M[X, Y]$ is a (vector-valued) functional, $f(X, Y) \equiv f(X(t), Y(t))$ is a (vector-valued) function of the state vector evaluated at time t . For all practical purposes, both $M[\cdot]$ and $f(\cdot)$ can be thought of as the measurement operator. The scalar Dirac delta function for time t_k is denoted with $\delta(t - t_k)$, and $\text{diag}\{\cdot\}$ stands for a diagonal matrix with the argument of $\text{diag}\{\cdot\}$ on the diagonal and zeros elsewhere. The formulation of (2.3) implies that the measurement Z_k has been taken at time t_k . Note that the t_k are not necessarily all different. Two measurements can be taken at the same time, for example as different pixels of the same remote sensing image.

But note that (2.3) constrains the set of possible measurements to point measurements in time, i.e. we cannot express measurements which integrate the state over time. For the soil moisture application, we will not need such measurements.

For the statistics of the process noise ν and ω , the measurement noise v , and the parameters α and β we assume

$$\begin{aligned}
\overline{\nu(t)} &\equiv 0 & \overline{\nu(t_1)\nu(t_2)^T} &= C_\nu(t_1, t_2) \\
\overline{\omega(t)} &\equiv 0 & \overline{\omega(t_1)\omega(t_2)^T} &= C_\omega(t_1, t_2) \\
\overline{v} &= 0 & \overline{vv^T} &= C_v \\
\overline{(\alpha - \bar{\alpha})} &= 0 & \overline{(\alpha - \bar{\alpha})(\alpha - \bar{\alpha})^T} &= C_\alpha \\
\overline{(\beta - \bar{\beta})} &= 0 & \overline{(\beta - \bar{\beta})(\beta - \bar{\beta})^T} &= C_\beta
\end{aligned} \tag{2.4}$$

where the overbar is the expectation operator. We generally denote the covariance of a random variable ξ with C_ξ . The superscript T denotes the matrix transpose. The prior values $\bar{\alpha}$ and $\bar{\beta}$ are our best guesses for the parameters prior to using the data Z . Moreover, we assume that all cross-covariances between ν , ω , v , α , and β vanish.

$$\begin{aligned}
\overline{\omega(t)\nu(t')^T} &\equiv 0 & \overline{v\nu(t)^T} &\equiv 0 & \overline{\alpha\nu(t)^T} &\equiv 0 & \overline{\beta\nu(t)^T} &\equiv 0 \\
\overline{v\omega(t)^T} &\equiv 0 & \overline{\alpha\omega(t)^T} &\equiv 0 & \overline{\beta\omega(t)^T} &\equiv 0 & & \\
\overline{v\alpha^T} &= 0 & \overline{v\beta^T} &= 0 & \overline{(\alpha - \bar{\alpha})(\beta - \bar{\beta})^T} &= 0 & &
\end{aligned} \tag{2.4a}$$

It is important to keep in mind that we only consider second-order statistics and implicitly assume distributions to be Gaussian or at least close to Gaussian. The computation of the reduced objective (Section 2.3.6) and of the posterior covariances (Section 2.4) will allow for tests of this crucial assumption.

Finally note that the measurement error covariance C_v can usually be made block-diagonal for remote sensing applications. In the soil moisture application, the snapshots of brightness temperature at different observation times are usually uncorrelated, although the measurement errors in each image are typically spatially correlated. Arranging the measurements within Z appropriately allows us to express the covariance C_v as a block-diagonal matrix, with the blocks containing the spatial correlation matrix of the measurement error.

2.1.2 Objective Function

Without loss of generality, we assume that there are no measurements exactly at the initial or the final time, that is we assume $t_k \in (0, t_f)$ and use $[0, t_f]$ as the time window for the inversion. The performance index for the estimation problem is the objective function

$$\begin{aligned}
\tilde{J} &= (Z - M[X, Y])^T C_v^{-1} (Z - M[X, Y]) \\
&+ (\alpha - \bar{\alpha})^T C_\alpha^{-1} (\alpha - \bar{\alpha}) + (\beta - \bar{\beta})^T C_\beta^{-1} (\beta - \bar{\beta}) \\
&+ \int_0^{t_f} \int_0^{t_f} \nu(t')^T C_\nu^{-1}(t', t'') \nu(t'') dt' dt'' + \int_0^{t_f} \int_0^{t_f} \omega(t')^T C_\omega^{-1}(t', t'') \omega(t'') dt' dt''
\end{aligned} \tag{2.5}$$

which will be minimized with respect to $\nu(t)$, $\omega(t)$, α , and β subject to the state equation (2.1) as a constraint. The first term accounts for the misfit between the data and the

model predictions of the measurements. The weights are given by the inverse covariance matrices of the measurement errors. The second and third terms penalize the deviation of the parameter vectors from their prior values $\bar{\alpha}$ and $\bar{\beta}$. The weights are given by the inverse covariance matrices C_α^{-1} and C_β^{-1} of the respective parameters.

Finally, the last two terms accounts for the model error. Note that the weight here is an inverse covariance function, which is defined by the operator identity

$$\int_0^{t_f} C_\xi(t, t') C_\xi^{-1}(t', t'') dt' = \delta(t - t'') I_{N_\xi} \quad (2.6)$$

where I_{N_ξ} is the $N_\xi \times N_\xi$ identity matrix and $\delta(\cdot)$ is again the Dirac delta function.

2.1.3 Euler-Lagrange Equations

In order to derive the Euler-Lagrange equations, we adjoin the state equation (2.1) and the initial condition (2.1a) as constraints to the objective function \tilde{J} (2.5). This step introduces the adjoint variables $\mu(t)$, a vector of length N_X , as well as $\lambda(t)$ and λ_0 , both vectors of length N_Y .

$$\begin{aligned} J = & \tilde{J} \\ & - 2 \int_0^{t_f} \mu^T (\phi(X, Y; \alpha) + D_\nu P_\nu \nu) dt \\ & + 2 \int_0^{t_f} \lambda^T \left(\frac{\partial Y}{\partial t} - \varphi(X, Y; \alpha) - D_\omega(Y) P_\omega \omega \right) dt + 2\lambda_0^T (Y|_{t=0} - Y_0(\beta)) \end{aligned} \quad (2.7)$$

The minimization of the objective function is a straightforward application of the calculus of variations [Courant and Hilbert, 1953; Lanczos, 1966]. Details of the derivation are outlined in Appendix A.1. We eventually get a set of equations for the estimates \hat{X} , \hat{Y} , $\hat{\nu}$, $\hat{\omega}$, $\hat{\alpha}$, and $\hat{\beta}$ of the state, the process noise, and the parameters, respectively. We call this set the *Euler-Lagrange equations*.

Forward Equation

$$0 = \phi(\hat{X}, \hat{Y}; \hat{\alpha}) + D_\nu P_\nu \hat{\nu}$$

$$\frac{\partial \hat{Y}}{\partial t} = \varphi(\hat{X}, \hat{Y}; \hat{\alpha}) + D_\omega(\hat{Y}) P_\omega \hat{\omega}$$

(2.8)

$$\hat{Y}|_{t=0} = Y_0(\hat{\beta})$$

(2.8a)

Backward Equation

$$\begin{aligned}
0 &= \widehat{\frac{\partial \phi}{\partial X}}^T \mu + \widehat{\frac{\partial \varphi}{\partial X}}^T \lambda + \widehat{\frac{\partial f}{\partial X}}^T [\delta] C_v^{-1} (Z - M[\hat{X}, \hat{Y}]) \\
-\frac{\partial \lambda}{\partial t} &= \widehat{\frac{\partial \phi}{\partial Y}}^T \mu + \widehat{\frac{\partial \varphi}{\partial Y}}^T \lambda + \frac{\partial [D_\omega(\hat{Y}) P_\omega \hat{\omega}]}{\partial Y}^T \lambda \\
&\quad + \widehat{\frac{\partial f}{\partial Y}}^T [\delta] C_v^{-1} (Z - M[\hat{X}, \hat{Y}])
\end{aligned}$$

(2.9)

$\lambda|_{t=t_f} = 0$

(2.9a)

Parameter Update

$$\begin{aligned}
\hat{\alpha} &= \bar{\alpha} + C_\alpha \int_0^{t_f} \left(\widehat{\frac{\partial \phi}{\partial \alpha}}^T \mu + \widehat{\frac{\partial \varphi}{\partial \alpha}}^T \lambda \right) dt \\
\hat{\beta} &= \bar{\beta} + C_\beta \widehat{\frac{\partial Y_0}{\partial \beta}}^T \lambda|_{t=0}
\end{aligned}$$

(2.10)

Noise Update

$$\begin{aligned}
\hat{\nu} &= \int_0^{t_f} C_\nu(t, t') P_\nu^T D_\nu^T \mu(t') dt' \\
\hat{\omega} &= \int_0^{t_f} C_\omega(t, t') P_\omega^T D_\omega(\hat{Y}(t'))^T \lambda(t') dt'
\end{aligned}$$

(2.11)

Note that all partial derivatives of ϕ , φ , f , and Y_0 are evaluated at the estimates \hat{X} , \hat{Y} , $\hat{\alpha}$, and $\hat{\beta}$, for which we use the shortcut notation

$$\widehat{\frac{\partial \phi}{\partial X}} \equiv \frac{\partial \phi}{\partial X} \Big|_{\hat{X}, \hat{Y}, \hat{\alpha}} \quad \widehat{\frac{\partial \varphi}{\partial \alpha}} \equiv \frac{\partial \varphi}{\partial \alpha} \Big|_{\hat{X}, \hat{Y}, \hat{\alpha}} \quad \widehat{\frac{\partial f}{\partial Y}} \equiv \frac{\partial f}{\partial Y} \Big|_{\hat{X}, \hat{Y}} \quad \text{etc.} \quad (2.12)$$

Equation (2.8) resembles the state equation and is solved forward in time subject to the initial condition (2.8a). This so-called *forward equation* uses the estimates of the parameters and is forced with the estimate of the process noise. The *parameter update* (2.10) and the *process noise update* (2.11) are in turn determined by the solution of the adjoint equation (2.9), which is subject to the terminal condition (2.9a). Since the adjoint equation is solved backward in time, we call it the *backward equation*. Note that the backward equation is linear in the adjoint variables μ and λ . At measurement times t_k , the backward equation is forced with the misfit between the data and the estimates of the observations. But to compute the latter, we need the state estimates from the forward equation (2.8).

Obviously, the Euler-Lagrange equations present a highly coupled, nonlinear set of equations. Notice that even for linear state and measurement equations the Euler-Lagrange equations are coupled through the data misfit term in the backward equation. This coupling leaves us with a two-point boundary value problem. Sections 2.2 and 2.3 provide details on how the Euler-Lagrange equations are solved.

2.2 Solving the Nonlinear Euler-Lagrange Equations: Iterated Representers

A simple solution strategy for the Euler-Lagrange equations is to iterate simultaneously on all the nonlinear terms as well as on the data misfit coupling term in the backward equation. Breaking the coupling in this way leaves us with two initial value problems at each iteration, which are easily integrated. However, this approach failed to converge for the soil moisture problem. Similar difficulties for oceanographic applications have been reported by Bennett [1992].

Alternatively, we can solve the nonlinear estimation problem as a sequence of linear estimation problems [Bennett, 1992]. Rather than linearizing all the coupling terms, we can linearize the Euler-Lagrange equations with a standard Taylor series expansion around a given trajectory and at the same time keep the coupling through the data misfit term in the backward equation. We are then left with having to solve a linear two-point boundary value problem in every iteration. Each of these two-point boundary value problems corresponds to solving the estimation problem for a set of linearized state and measurement equations. Equivalently, if we consistently linearize the state and measurement equations with a standard Taylor series expansion around the given trajectory, and if we then derive the estimation equations for these linearized state and measurement equations, we find that they are exactly the linearized Euler-Lagrange equations. Below we outline this approach in detail.

An elegant technique for solving a linear Euler-Lagrange system is the representer approach [Bennett, 1992; Bennett, 1999]. In this approach the Euler-Lagrange equations are decoupled by introducing a series expansion solution of the estimate around so-called representer fields. The representer fields are the prior cross-covariances of the measurement predictions with the state, which implies that there are as many representer fields as there are measurements. Fortunately, we do not have to compute every representer field in order to get the best estimate. It is sufficient to compute a sequence of suitable linear combinations of the representer fields. This approach is called the indirect representer method, and we describe the technique in detail in Section 2.3. We use the indirect representer method in combination with the iteration on the nonlinearity to obtain the estimates \hat{X} , \hat{Y} , $\hat{\alpha}$, and $\hat{\beta}$. After the estimates have been computed, we can use the representer method to compute some posterior covariances (Section 2.4). A summary of the algorithm can be found in Figure 2.1 and in Section 2.3.5.

2.2.1 Tangent-linear Model

In meteorology and oceanography the state equation linearized around a trajectory is called the tangent-linear model. Our approach for solving the nonlinear Euler-Lagrange equations (2.8)–(2.11) is based on successive tangent-linearizations of the state and measurement equations. In this Section, we outline the linearization procedure.

Let X^η , Y^η , ω^η , α^η , and β^η denote the best estimates of the previous iteration level η . At the current iteration level $\eta + 1$, we linearize the state and measurement equations around the trajectories $X^\eta(t)$, $Y^\eta(t)$, and $\omega^\eta(t)$ as well as around the previous parameter

estimates α^η and β^η . Simplifying the linearization in ω^η according to

$$\begin{aligned} D_\omega(Y)P_\omega\omega &\approx D_\omega(Y^\eta)P_\omega\omega^\eta + \left. \frac{\partial[D_\omega(Y)P_\omega\omega]}{\partial Y} \right|_\eta (\omega - \omega^\eta) + \left. \frac{\partial[D_\omega(Y)P_\omega\omega]}{\partial Y} \right|_\eta (Y - Y^\eta) \\ &\equiv D_\omega(Y^\eta)P_\omega\omega + \left. \frac{\partial[D_\omega(Y)P_\omega\omega]}{\partial Y} \right|_\eta (Y - Y^\eta) \end{aligned}$$

and neglecting terms of second and higher order, we obtain the *tangent-linear state equation*

$$\begin{aligned} 0 &= \phi(X^\eta, Y^\eta; \alpha^\eta) + \left. \frac{\partial\phi}{\partial X} \right|_\eta (X - X^\eta) + \left. \frac{\partial\phi}{\partial Y} \right|_\eta (Y - Y^\eta) + \left. \frac{\partial\phi}{\partial\alpha} \right|_\eta (\alpha - \alpha^\eta) \\ &\quad + D_\nu P_\nu \nu \\ \frac{\partial Y}{\partial t} &= \varphi(X^\eta, Y^\eta; \alpha^\eta) + \left. \frac{\partial\varphi}{\partial X} \right|_\eta (X - X^\eta) + \left. \frac{\partial\varphi}{\partial Y} \right|_\eta (Y - Y^\eta) + \left. \frac{\partial\varphi}{\partial\alpha} \right|_\eta (\alpha - \alpha^\eta) \\ &\quad + D_\omega(Y^\eta)P_\omega\omega + \left. \frac{\partial[D_\omega(Y)P_\omega\omega]}{\partial Y} \right|_\eta (Y - Y^\eta) \end{aligned} \quad (2.13)$$

$$Y|_{t=0} = Y_0(\beta^\eta) + \left. \frac{\partial Y_0}{\partial\beta} \right|_\eta (\beta - \beta^\eta) \quad (2.13a)$$

The *tangent-linear measurement equation* reads

$$\begin{aligned} Z &= M[X^\eta, Y^\eta] + L[X - X^\eta, Y - Y^\eta] + v \\ L_k[X, Y] &\equiv \int_0^{t_f} \left\{ \left. \frac{\partial f_k}{\partial X} \right|_\eta X(t) + \left. \frac{\partial f_k}{\partial Y} \right|_\eta Y(t) \right\} \delta(t - t_k) dt \\ &\equiv \left. \frac{\partial f_k}{\partial X} \right|_{\eta, t_k} X(t_k) + \left. \frac{\partial f_k}{\partial Y} \right|_{\eta, t_k} Y(t_k) \end{aligned} \quad (2.14)$$

We used (2.3) to get this explicit expression for the linear (vector-valued) functional $L[\cdot]$, which can be interpreted as the slope of the measurement operator. For the partial derivatives we use the obvious short-cut notation

$$\left. \frac{\partial\phi}{\partial X} \right|_\eta \equiv \left. \frac{\partial\phi}{\partial X} \right|_{X^\eta, Y^\eta; \alpha^\eta} \quad \left. \frac{\partial\varphi}{\partial\alpha} \right|_\eta \equiv \left. \frac{\partial\varphi}{\partial\alpha} \right|_{X^\eta, Y^\eta; \alpha^\eta} \quad \left. \frac{\partial f}{\partial Y} \right|_\eta \equiv \left. \frac{\partial f}{\partial Y} \right|_{X^\eta, Y^\eta} \quad \text{etc.}$$

Note that the linear operator $L[\cdot]$ of the tangent-linear measurement equation changes with each iteration. In particular, $L[\cdot]$ of the current iteration level $\eta + 1$ is obtained by evaluating the partial derivative of $f(\cdot)$ at the best estimate of the previous iteration level. Consequently, $L[\cdot]$ should carry a superscript $\eta + 1$. However, we opt to drop this superscript because it is easy to infer from the context.

2.2.2 Linearized Euler-Lagrange Equations

We obtain a “linearized” version of the objective function by introducing the linearizations of (2.13) and (2.14) into (2.7). Following the general procedure outlined in Appendix A.1 for the derivation of the nonlinear Euler-Lagrange equations, we can easily derive the linearized

Euler-Lagrange equations from the “linearized” objective function.

$$\begin{aligned}
0 &= \left. \frac{\partial \phi}{\partial X} \right|_{\eta}^T \mu^{\eta+1} + \left. \frac{\partial \varphi}{\partial X} \right|_{\eta}^T \lambda^{\eta+1} \\
&\quad + \left. \frac{\partial f}{\partial X} \right|_{\eta}^T [\delta] C_v^{-1} (Z - M[X^{\eta}, Y^{\eta}] - L[X^{\eta+1} - X^{\eta}, Y^{\eta+1} - Y^{\eta}]) \\
-\frac{\partial \lambda^{\eta+1}}{\partial t} &= \left. \frac{\partial \phi}{\partial Y} \right|_{\eta}^T \mu^{\eta+1} + \left. \frac{\partial \varphi}{\partial Y} \right|_{\eta}^T \lambda^{\eta+1} + \left. \frac{\partial [D_{\omega}(Y) P_{\omega} \omega]}{\partial Y} \right|_{\eta}^T \lambda^{\eta+1}
\end{aligned} \tag{2.15}$$

$$\begin{aligned}
&\quad + \left. \frac{\partial f}{\partial Y} \right|_{\eta}^T [\delta] C_v^{-1} (Z - M[X^{\eta}, Y^{\eta}] - L[X^{\eta+1} - X^{\eta}, Y^{\eta+1} - Y^{\eta}]) \\
\lambda|_{t=t_f}^{\eta+1} &= 0
\end{aligned} \tag{2.15a}$$

$$\alpha^{\eta+1} = \bar{\alpha} + C_{\alpha} \int_0^{t_f} \left(\left. \frac{\partial \phi}{\partial \alpha} \right|_{\eta}^T \mu^{\eta+1} + \left. \frac{\partial \varphi}{\partial \alpha} \right|_{\eta}^T \lambda^{\eta+1} \right) dt \tag{2.16}$$

$$\beta^{\eta+1} = \bar{\beta} + C_{\beta} \left. \frac{\partial Y_0}{\partial \beta} \right|_{\eta}^T \lambda|_{t=0}^{\eta+1}$$

$$\begin{aligned}
\nu^{\eta+1} &= \int_0^{t_f} C_{\nu}(t, t') P_{\nu}^T D_{\nu}^T \mu^{\eta+1}(t') dt' \\
\omega^{\eta+1} &= \int_0^{t_f} C_{\omega}(t, t') P_{\omega}^T [D_{\omega}(Y^{\eta}(t'))]^T \lambda^{\eta+1}(t') dt'
\end{aligned} \tag{2.17}$$

$$\begin{aligned}
0 &= \phi(X^{\eta}, Y^{\eta}; \alpha^{\eta}) + \left. \frac{\partial \phi}{\partial X} \right|_{\eta} (X^{\eta+1} - X^{\eta}) + \left. \frac{\partial \phi}{\partial Y} \right|_{\eta} (Y^{\eta+1} - Y^{\eta}) + \left. \frac{\partial \phi}{\partial \alpha} \right|_{\eta} (\alpha^{\eta+1} - \alpha^{\eta}) \\
&\quad + D_{\nu} P_{\nu} \nu^{\eta+1} \\
\frac{\partial Y^{\eta+1}}{\partial t} &= \varphi(X^{\eta}, Y^{\eta}; \alpha^{\eta}) + \left. \frac{\partial \varphi}{\partial X} \right|_{\eta} (X^{\eta+1} - X^{\eta}) + \left. \frac{\partial \varphi}{\partial Y} \right|_{\eta} (Y^{\eta+1} - Y^{\eta}) + \left. \frac{\partial \varphi}{\partial \alpha} \right|_{\eta} (\alpha^{\eta+1} - \alpha^{\eta}) \\
&\quad + D_{\omega}(Y^{\eta}) P_{\omega} \omega^{\eta+1} + \left. \frac{\partial [D_{\omega}(Y) P_{\omega} \omega]}{\partial Y} \right|_{\eta} (Y^{\eta+1} - Y^{\eta})
\end{aligned} \tag{2.18}$$

$$Y|_{t=0}^{\eta+1} = Y_0(\beta^{\eta}) + \left. \frac{\partial Y_0}{\partial \beta} \right|_{\eta} (\beta^{\eta+1} - \beta^{\eta}) \tag{2.18a}$$

Having completed iteration level η , we need to solve these linearized Euler-Lagrange equations for the new estimates $\mu^{\eta+1}$, $\lambda^{\eta+1}$, $\alpha^{\eta+1}$, $\beta^{\eta+1}$, $\nu^{\eta+1}$, $\omega^{\eta+1}$, $X^{\eta+1}$ and $Y^{\eta+1}$. The iteration can be initialized by setting the initial best estimates X^{η} and Y^{η} for $\eta = 0$ equal to the prior trajectories \bar{X} and \bar{Y} of the nonlinear model, which are the solution of the

nonlinear Euler-Lagrange equations when no data are available. In particular, \bar{X} and \bar{Y} are defined through

$$0 = \phi(\bar{X}, \bar{Y}; \bar{\alpha}) \quad \frac{\partial \bar{Y}}{\partial t} = \varphi(\bar{X}, \bar{Y}; \bar{\alpha}) \quad \bar{Y}|_{t=0} = Y_0(\bar{\beta}) \quad (2.19)$$

Note that $\bar{\mu} \equiv \bar{\lambda} \equiv 0$.

As mentioned in the introduction to this Section, the linearized Euler-Lagrange equations (2.15)–(2.18) can also be obtained by linearizing the nonlinear estimation equations (2.8)–(2.11). However, care must be taken that the linearization be carried out consistently, because there is an infinite number of ways in which the Euler-Lagrange equations can be linearized. Re-deriving the linearized Euler-Lagrange equations from the tangent-linear model ensures that the meaning of the linearized Euler-Lagrange equations as the iterated estimation equations of the original problem is preserved.

Although there is no proof or guarantee that the sequence defined above converges, experience has shown that it converges in practice for forward models that are reasonably close to linear. For a quasigeostrophic model, Bennett [1992] provides a theorem on convergence in a doubly-periodic domain. The sequence is then bounded, and so must have points of accumulation or cluster points, but not necessarily unique limits.

Solving the nonlinear problem as a series of linear estimation problems is clearly motivated by the fact that we are certain to obtain a solution to the linear estimation problem at each iterate. Before implementing the full estimation technique, it is therefore a good idea to check whether the tangent-linear model converges to the solution of the nonlinear forward model, which is equivalent to solving the estimation problem with no data. Without fulfilling this prerequisite there is little hope that the sequence will converge when data are assimilated.

2.3 Solving the Linear Euler-Lagrange Equations: Indirect Representers

The focus of this Section is to illustrate the representer solution of the *linearized* Euler-Lagrange equations (2.15)–(2.18). The representer approach is a very elegant way to decouple and solve the linear Euler-Lagrange equations. The approach is based on series (or representer) expansions for the estimates of the state and its adjoint variable. The series expansions linearly superimpose the measurement updates from each individual observation, implying that the representer solution is inherently the solution to a *linear* estimation problem.

In essence, the representer solution reduces the size of the space in which the objective is minimized from infinity to a finite dimension equal to the number of observations. This promises better convergence behavior than a direct minimization of the objective function with a gradient search. Fortunately, the *indirect representer method* allows us to solve for the estimates without explicitly computing all of the individual representer fields, that is we do not necessarily have to solve the basic equation (forward or backward) $2N_Z$ times (per nonlinear iteration). The indirect representer method therefore provides us with a fast way to get the estimate [Bennett et al., 1996]. If we opt for computing the individual representer fields after the nonlinear iteration has converged and the estimate has been obtained, the representer approach also yields a lot of information about the posterior error covariance (Section 2.4).

We conclude this Section with a short discussion of the hypothesis test associated with the value of the reduced objective function (Section 2.3.6) and a summary of the iterated indirect representer algorithm (Section 2.3.5).

2.3.1 Representer Expansion

The representer approach is essentially a series expansion solution of the estimation problem, where the number of unknown coefficients exactly matches the number of available measurements. For the new estimates of the current iteration level $\eta + 1$, we define

$$X^{\eta+1}(t) = \bar{X}^{\eta+1}(t) + \sum_{k=1}^{N_Z} b_k \Xi^k(t) \quad Y^{\eta+1}(t) = \bar{Y}^{\eta+1}(t) + \sum_{k=1}^{N_Z} b_k \Upsilon^k(t) \quad (2.20)$$

$$\mu^{\eta+1}(t) = \bar{\mu}^{\eta+1}(t) + \sum_{k=1}^{N_Z} b_k \Omega^k(t) \quad \lambda^{\eta+1}(t) = \bar{\lambda}^{\eta+1}(t) + \sum_{k=1}^{N_Z} b_k \Lambda^k(t) \quad (2.21)$$

The $4N_Z$ representer functions $\Xi^k(t)$, $\Upsilon^k(t)$, $\Omega^k(t)$ and $\Lambda^k(t)$ for the state and its adjoint are time-dependent vectors of length N_X and N_Y , respectively. We write Υ_i^k in order to refer to the i -th component of the k -th representer function Υ^k . The N_Z scalar representer coefficients b_k are constant in time. Note that the coefficients are the same for both the state and the adjoint representers.

The expansion is carried out around the prior fields $\bar{X}^{\eta+1}$, $\bar{Y}^{\eta+1}$, $\bar{\mu}^{\eta+1}$, and $\bar{\lambda}^{\eta+1}$, which are the solutions of the linearized Euler-Lagrange equations (2.15)–(2.18) when no data are available. In particular, $\bar{X}^{\eta+1}$ and $\bar{Y}^{\eta+1}$ are defined through

$$\begin{aligned} 0 = & \phi(X^\eta, Y^\eta; \alpha^\eta) + \left. \frac{\partial \phi}{\partial X} \right|_\eta (\bar{X}^{\eta+1} - X^\eta) + \left. \frac{\partial \phi}{\partial Y} \right|_\eta (\bar{Y}^{\eta+1} - Y^\eta) + \left. \frac{\partial \phi}{\partial \alpha} \right|_\eta (\bar{\alpha} - \alpha^\eta) \\ \frac{\partial \bar{Y}^{\eta+1}}{\partial t} = & \varphi(X^\eta, Y^\eta; \alpha^\eta) + \left. \frac{\partial \varphi}{\partial X} \right|_\eta (\bar{X}^{\eta+1} - X^\eta) + \left. \frac{\partial \varphi}{\partial Y} \right|_\eta (\bar{Y}^{\eta+1} - Y^\eta) + \left. \frac{\partial \varphi}{\partial \alpha} \right|_\eta (\bar{\alpha} - \alpha^\eta) \\ & + \left. \frac{\partial [D_\omega(Y) P_\omega \omega]}{\partial Y} \right|_\eta (\bar{Y}^{\eta+1} - Y^\eta) \end{aligned} \quad (2.22)$$

$$\bar{Y}|_{t=0}^{\eta+1} = Y_0(\beta^\eta) + \left. \frac{\partial Y_0}{\partial \beta} \right|_\eta (\bar{\beta} - \beta^\eta) \quad (2.22a)$$

and $\bar{\mu}^{\eta+1} \equiv \bar{\lambda}^{\eta+1} \equiv 0$. Recall that X^η and Y^η are the best estimates from the previous iteration and serve as the trajectories around which we linearize.

It is easy to see that the prior fields for iteration level $\eta + 1$ as defined above are the tangent-linear approximations of the prior trajectory (2.19) of the nonlinear problem. Obviously, one cannot simultaneously linearize around the previous estimate and the nonlinear prior trajectory (2.19). In order to achieve a dynamically consistent estimate, we must linearize around the previous best estimate, and the accuracy of the prior must be compromised. Therefore the nonlinear estimate cannot be strictly optimal. However, if the model is reasonably linear, the error so introduced is bearable.

On the Notation

Please note again that the superscript k that is used for the representer fields stands for the number of the measurement that corresponds to the representer function in question. This superscript is an integral part of the name of the representer function. Individual vector components of a given representer field are denoted with subscripts, usually i or j . For instance, Ξ_i^k denotes the i -th component of the representer field that corresponds to the k -th measurement. In contrast, the representer coefficients carry a subscript k . This subscript represents the number of the measurement to which the representer coefficient corresponds. It also serves as the index of the vector component when the representer coefficients b_k are collected into the vector b , in analogy to assembling the individual measurements in the data vector Z .

Moreover, it is important to note that the representer fields and the representer coefficients like the estimates and the prior trajectories of the states change with every iteration. Consequently, b_k , Ξ^k , Υ^k , Ω^k and Λ^k should have an additional superscript $\eta + 1$. Again, we opt to drop this superscript because it is easy to infer from the context. Unlike for the states, we never have to use representer fields or coefficients from different iteration levels in the same equation. Finally note that the priors of the original uncertain parameters $\bar{\alpha}$, $\bar{\beta}$, and $\bar{\nu}(t) \equiv \bar{\omega}(t) \equiv 0$ never change!

2.3.2 Representer Equations

By inserting the adjoint representer expansion (2.21) into the backward equation (2.15), and choosing

$$b = C_v^{-1} (Z - M[X^\eta, Y^\eta] - L[X^{\eta+1} - X^\eta, Y^{\eta+1} - Y^\eta]) \quad (2.23)$$

we can derive an equation for the adjoint representers.

Adjoint Representer Equations

$$0 = \frac{\partial \phi}{\partial X} \Big|_\eta^T \Omega^k + \frac{\partial \varphi}{\partial X} \Big|_\eta^T \Lambda^k + \delta(t - t_k) \frac{\partial f_k}{\partial X} \Big|_\eta^T \quad (2.24)$$

$$-\frac{\partial \Lambda^k}{\partial t} = \frac{\partial \phi}{\partial Y} \Big|_\eta^T \Omega^k + \frac{\partial \varphi}{\partial Y} \Big|_\eta^T \Lambda^k + \frac{\partial [D_\omega(Y) P_\omega \omega]}{\partial Y} \Big|_\eta^T \Lambda^k + \delta(t - t_k) \frac{\partial f_k}{\partial Y} \Big|_\eta^T$$

$\Lambda^k|_{t=t_f} = 0$

(2.24a)

Recall that the k -th measurement is taken at time t_k . If we compare the adjoint representer equation (2.24) with the backward equation (2.15), we can see that the two equations are very similar. Whereas the backward equation is forced with the posterior data misfit, the adjoint representer equation is forced with a single unit impulse at one measurement time and location. We can therefore solve for the adjoint representer fields without having to know the estimate in advance.

Next, we derive an equation for the state representers by inserting the parameter update (2.16), the process noise update (2.17), and the representer expansions (2.20) and (2.21) into the forward equation (2.18) and its initial condition (2.18a). Using the definition of

the prior fields (2.22), we find

State Representer Equations

$$\begin{aligned}
0 &= \frac{\partial \phi}{\partial X} \Big|_{\eta} \Xi^k + \frac{\partial \phi}{\partial Y} \Big|_{\eta} \Upsilon^k + \int_0^{t_f} D_{\nu} P_{\nu} C_{\nu}(t, t') P_{\nu}^T D_{\nu}^T \Omega^k(t') dt' \\
&\quad + \frac{\partial \phi}{\partial \alpha} \Big|_{\eta} C_{\alpha} \int_0^{t_f} \left(\frac{\partial \phi}{\partial \alpha} \Big|_{\eta}^T \Omega^k(t') + \frac{\partial \varphi}{\partial \alpha} \Big|_{\eta}^T \Lambda^k(t') \right) dt' \\
\frac{\partial \Upsilon^k}{\partial t} &= \frac{\partial \varphi}{\partial X} \Big|_{\eta} \Xi^k + \frac{\partial \varphi}{\partial Y} \Big|_{\eta} \Upsilon^k + \int_0^{t_f} D_{\omega}(Y^{\eta}(t)) P_{\omega} C_{\omega}(t, t') P_{\omega}^T D_{\omega}(Y^{\eta}(t'))^T \Lambda^k(t') dt' \\
&\quad + \frac{\partial \varphi}{\partial \alpha} \Big|_{\eta} C_{\alpha} \int_0^{t_f} \left(\frac{\partial \phi}{\partial \alpha} \Big|_{\eta}^T \Omega^k(t') + \frac{\partial \varphi}{\partial \alpha} \Big|_{\eta}^T \Lambda^k(t') \right) dt' \\
&\quad + \frac{\partial [D_{\omega}(Y) P_{\omega} \omega]}{\partial Y} \Big|_{\eta} \Upsilon^k
\end{aligned}
\tag{2.25}$$

$$\Upsilon^k|_{t=0} = \frac{\partial Y_0}{\partial \beta} \Big|_{\eta} C_{\beta} \frac{\partial Y_0}{\partial \beta} \Big|_{\eta}^T \Lambda^k|_{t=0}$$

(2.25a)

As will be shown in Sections 2.4.1 and A.2.1, the state representer function for the k -th measurement is equal to the prior cross-covariance of the model prediction for the k -th measurement and the states at all times and locations. Most importantly, the representer fields encapsulate only prior information and do not depend on the data. Finally note that we can solve for the state representer as soon as we have integrated the adjoint representer fields.

2.3.3 Representer Coefficients and Representer Matrix

In order to derive the representer equations, we defined the representer coefficients as

$$b = C_v^{-1} (Z - M[X^{\eta}, Y^{\eta}] - L[X^{\eta+1} - X^{\eta}, Y^{\eta+1} - Y^{\eta}]) \tag{2.23}$$

Using the representer expansion (2.20) for the estimate and the linearity of L , we can easily rewrite (2.23) as

$$b = U^{-1} (Z - M[X^{\eta}, Y^{\eta}] - L[\bar{X}^{\eta+1} - X^{\eta}, \bar{Y}^{\eta+1} - Y^{\eta}]) \tag{2.26}$$

where we define

$$U \equiv C_v + R \quad \text{and} \quad [R]_{kl} \equiv L_k[\Xi^l, \Upsilon^l] \tag{2.27}$$

Both U and the *representer matrix* R are symmetric $N_Z \times N_Z$ matrices. We can calculate the entries of the representer matrix (and hence U) if we compute all N_Z individual representer fields Ξ^k and Υ^k , and subsequently apply the operator L of (2.14).

Obviously, knowledge of the representer matrix suffices to compute the estimate. If we know the representer matrix, we can solve (2.26) for the representer coefficients. Once we have the representer coefficients, we can use the definition (2.23) and integrate the backward equation (2.15). After the backward integration, we can easily calculate the parameter update (2.16) and the noise update (2.17), and finally solve the forward equation (2.18) for the state estimates. The coupling between the forward and the backward equations of the original two-point boundary value problem (2.15)–(2.18) has now been broken by the representer. The price we pay for breaking the coupling is of course the increased computational burden.

The representer matrix encapsulates all the relevant prior information. In contrast to (2.23), Equation (2.26) expresses the representer coefficients entirely in prior terms, except for the explicit appearance of the data vector Z . For a linear estimation problem we could even precompute the representer matrix before we know the data, and then solve for the representer coefficients when the data become available. Solving for the optimal estimates via explicit computation of the representer matrix is called the direct representer method.

2.3.4 Indirect Representers

In particular for data-rich remote sensing applications, the computational burden for the calculation of the representer matrix is very heavy, because all N_Z individual representer fields must be integrated. But recall that we really only need the representer coefficients to compute the estimated fields. Once we have the representer coefficients, we can integrate the backward equation (2.15) and subsequently the forward equation (2.18). Fortunately, we do not need to know the representer matrix explicitly in order to get the representer coefficients. The conjugate gradient technique for solving linear equations like (2.26) does the trick.

A conjugate gradient solver [Press et al., 1992] successively approximates the solution of a linear equation. In our case, this means constructing successive approximations of the representer coefficient vector b . Let us rewrite (2.26) as $Ub = \text{rhs}$, where rhs is obviously the known prior data misfit. Given an approximation ζ of the representer coefficient vector b , the conjugate gradient technique only uses the product $U\zeta$ to refine the approximation. If we can supply the conjugate gradient algorithm with the product $U\zeta$ for any given vector ζ , we will eventually find the representer coefficients. We never need to supply the matrix U itself to the conjugate gradient solver.

It turns out that we can compute the product $U\zeta$ without explicit knowledge of the matrix U (or equivalently R) [Bennett et al., 1996]. All we need to do is solve the representer equations for a linear combination of inhomogeneities. For a given vector ζ , we solve the following equations for the fields Ξ and Υ .

$$\begin{aligned} 0 &= \left. \frac{\partial \phi}{\partial X} \right|_{\eta}^T \Omega + \left. \frac{\partial \varphi}{\partial X} \right|_{\eta}^T \Lambda + \sum_{k=1}^{N_z} \delta(t - t_k) \left. \frac{\partial f_k}{\partial X} \right|_{\eta}^T \zeta_k \\ -\frac{\partial \Lambda}{\partial t} &= \left. \frac{\partial \phi}{\partial Y} \right|_{\eta}^T \Omega + \left. \frac{\partial \varphi}{\partial Y} \right|_{\eta}^T \Lambda + \left. \frac{\partial [D_{\omega}(Y) P_{\omega} \omega]}{\partial Y} \right|_{\eta}^T \Lambda + \sum_{k=1}^{N_z} \delta(t - t_k) \left. \frac{\partial f_k}{\partial Y} \right|_{\eta}^T \zeta_k \\ \Lambda|_{t=t_f} &= 0 \end{aligned}$$

$$\begin{aligned}
0 &= \frac{\partial \phi}{\partial X} \Big|_{\eta} \Xi + \frac{\partial \phi}{\partial Y} \Big|_{\eta} \Upsilon + \int_0^{t_f} D_{\nu} P_{\nu} C_{\nu}(t, t') P_{\nu}^T D_{\nu}^T \Omega(t') dt' \\
&\quad + \frac{\partial \phi}{\partial \alpha} \Big|_{\eta} C_{\alpha} \int_0^{t_f} \left(\frac{\partial \phi}{\partial \alpha} \Big|_{\eta}^T \Omega(t') + \frac{\partial \varphi}{\partial \alpha} \Big|_{\eta}^T \Lambda(t') \right) dt' \\
\frac{\partial \Upsilon}{\partial t} &= \frac{\partial \varphi}{\partial X} \Big|_{\eta} \Xi + \frac{\partial \varphi}{\partial Y} \Big|_{\eta} \Upsilon + \int_0^{t_f} D_{\omega}(Y^{\eta}(t)) P_{\omega} C_{\omega}(t, t') P_{\omega}^T D_{\omega}(Y^{\eta}(t'))^T \Lambda(t') dt' \\
&\quad + \frac{\partial \varphi}{\partial \alpha} \Big|_{\eta} C_{\alpha} \int_0^{t_f} \left(\frac{\partial \phi}{\partial \alpha} \Big|_{\eta}^T \Omega(t') + \frac{\partial \varphi}{\partial \alpha} \Big|_{\eta}^T \Lambda(t') \right) dt' + \frac{\partial [D_{\omega}(Y) P_{\omega} \omega]}{\partial Y} \Big|_{\eta} \Upsilon \\
\Upsilon|_{t=0} &= \frac{\partial Y_0}{\partial \beta} \Big|_{\eta} C_{\beta} \frac{\partial Y_0}{\partial \beta} \Big|_{\eta}^T \Lambda|_{t=0}
\end{aligned}$$

Having computed the fields Ξ and Υ , we have by way of linearity

$$R\zeta = L[\Xi, \Upsilon]$$

It is now simple to compute $R\zeta + C_{\nu}\zeta \equiv U\zeta$, which is all we need to supply to the conjugate gradient solver. Note that in the indirect representer approach we are effectively computing linear combinations of the representer fields, namely $\Xi = \sum_k \zeta_k \Xi^k$ and $\Upsilon = \sum_k \zeta_k \Upsilon^k$.

Numerical issues aside, the conjugate gradient solver for our linear system is guaranteed to converge after at most N_Z iterations, that is we need to compute at most N_Z linear combinations of representer fields. This means that we can only do better compared with the direct approach of assembling the representer matrix, for which we need exactly N_Z (individual) representer fields. The actual number of iterations for the conjugate gradient solver depends of course on the problem at hand. For the land surface application, the savings are substantial. Detailed results on the efficiency of the indirect representer method are presented in Section 8.2.1.

Finally, preconditioning with an approximation of U (and hence the representer matrix R) can further improve the convergence speed [Press et al., 1992]. Bennett [1999] discusses several options for a preconditioner. If only one data type is assimilated, preconditioning does not seem necessary at all.

2.3.5 Summary of the Iterated Indirect Representer Algorithm

Figure 2.1 shows a flowchart summarizing the iterated indirect representer algorithm. For simplicity, we omit the diagnostic state vector X in the Figure and in the discussion. We start by computing the prior state trajectory \bar{Y} from the nonlinear state equation with the model errors and the uncertain parameters set to their prior values (2.19). Next, we initialize the previous best estimate of the first iteration as $Y^0 \equiv \bar{Y}$ and enter the loop of iterations on the nonlinearity in the state equation. The iteration on the nonlinearity is the outer loop of the algorithm.

During each iteration, the state and measurement equations are linearized around the trajectory of the previous best estimate Y^{η} (2.13)–(2.14). First, we compute the prior trajectory $\bar{Y}^{\eta+1}$ of the current iteration (2.22). From this prior trajectory, we derive the

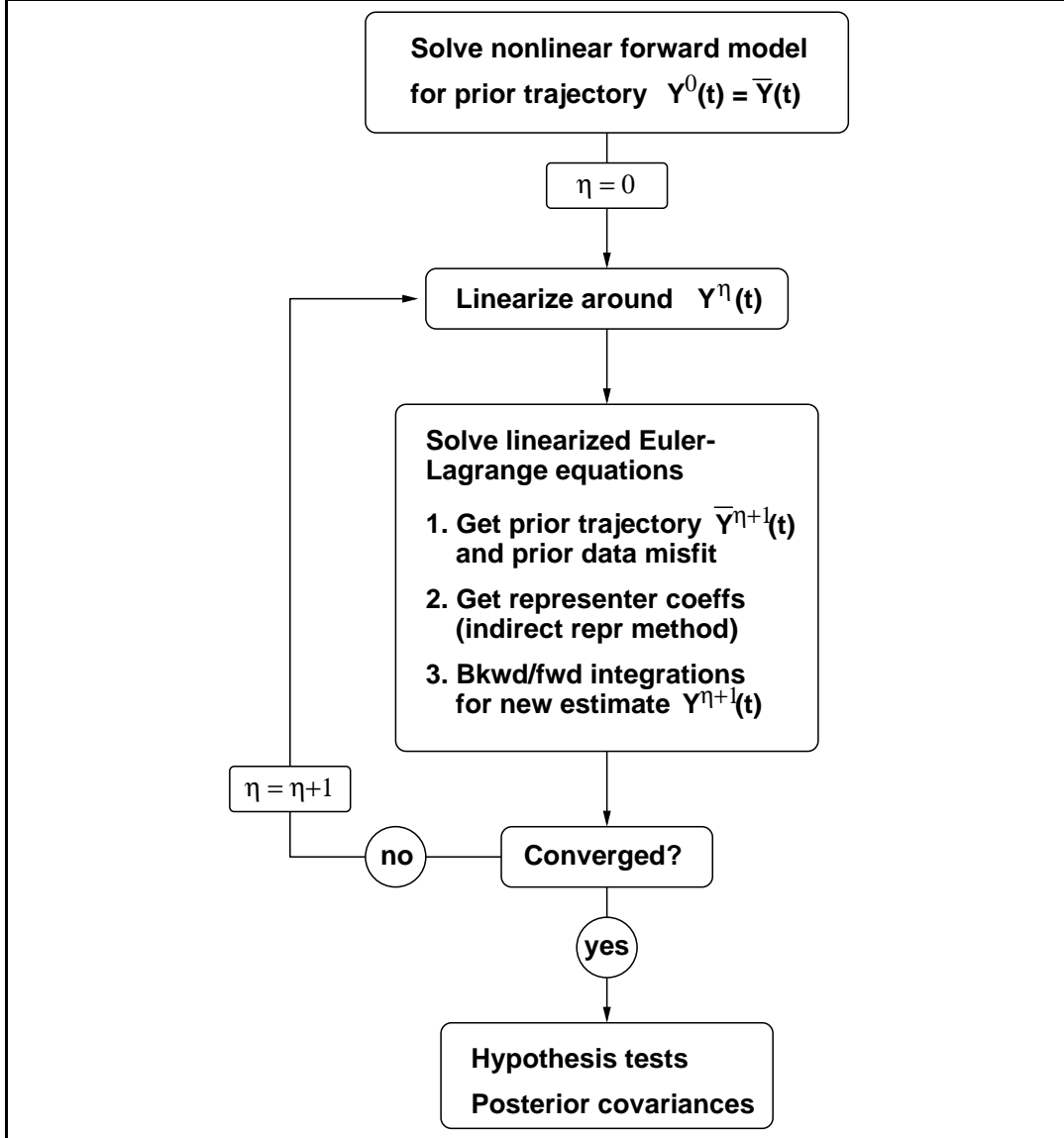


Figure 2.1: Flowchart for the iterated indirect representer method. A summary of the algorithm is presented in Section 2.3.5. The diagnostic state X has been omitted for clarity.

corresponding prior data misfit. Next, we calculate the representer coefficients of the current iteration with the indirect representer method (Section 2.3.4). Note that this constitutes an inner loop in which we successively refine our best estimate of the representer coefficients with a conjugate gradient algorithm. During each of the inner iterations, we essentially compute one linear combination of representer fields. After having obtained the representer coefficients, we finally compute the new estimate $Y^{\eta+1}$ using the definition of the representer coefficients (2.23) in the linearized Euler-Lagrange equations (2.15)–(2.18), which are then decoupled.

The convergence check is carried out by comparing the previous best estimate Y^η to the current best estimate $Y^{\eta+1}$. In particular, we check whether $\|Y^\eta - Y^{\eta+1}\| < \epsilon_Y$ for a given threshold ϵ_Y . Additional convergence checks are done for the uncertain parameters, for instance $\|\beta^\eta - \beta^{\eta+1}\| < \epsilon_\beta$. After convergence has been achieved, we may compute posterior covariances (Section 2.4) and carry out hypothesis tests on the reduced objective function (Section 2.3.6) or on the posterior data residuals (Section 2.4.1). Note that the hypothesis test on the reduced objective function does not require any posterior covariance calculations and can be done at negligible cost.

2.3.6 Reduced Objective Function

The objective function at the minimum is given by

$$\begin{aligned} \hat{J}^{\eta+1} &= (Z - M[X^\eta, Y^\eta] - L[\bar{X}^{\eta+1} - X^\eta, \bar{Y}^{\eta+1} - Y^\eta])^T U^{-1} \cdot \\ &\quad \cdot (Z - M[X^\eta, Y^\eta] - L[\bar{X}^{\eta+1} - X^\eta, \bar{Y}^{\eta+1} - Y^\eta]) \\ &= (Z - M[X^\eta, Y^\eta] - L[\bar{X}^{\eta+1} - X^\eta, \bar{Y}^{\eta+1} - Y^\eta])^T b \end{aligned} \quad (2.28)$$

and can be evaluated at negligible cost once the representer coefficients are known [Bennett, 1992]. If our assumptions about the model physics and the prior statistics are true, and if we invoke the central limit theorem in case we do not claim that the priors are exactly normally distributed, then the reduced objective is a chi-squared random variable with N_Z degrees of freedom

$$\hat{J}^{\eta+1} \sim \chi_{N_Z}^2 \quad \overline{\chi_{N_Z}^2} = N_Z \quad \text{var}(\chi_{N_Z}^2) = 2N_Z \quad (2.29)$$

We can therefore formulate our assumptions about the model physics (2.1), the measurement process (2.2), and the prior statistics (2.4) as a null hypothesis and apply the chi-square test to see whether these assumptions are statistically consistent with the data. Should our null hypothesis fail the test, we are forced to discard the estimated fields. In this case, however, we would have learned something about land surface dynamics.

Note that the hypothesis test, much like the estimates, relies on the successive tangent-linearizations. In the case of nonlinear model dynamics, the test can only be valid approximately.

2.4 Posterior Covariances

After the nonlinear iteration has converged, we can opt for calculating the posterior covariances of the state and the observations. These may be obtained by computing the individual representer functions Ξ^k and Υ^k (rather than the linear combinations Ξ and Υ

of the indirect representer approach) [Bennett, 1992]. This Section provides the expressions for the posterior covariances as functions of the representer.

Since the problem at hand is nonlinear, the linearized posterior covariances derived in this Section can at best be approximations of the true posterior covariances. For example, the prior fields $\bar{X}^{\eta+1}$ and $\bar{Y}^{\eta+1}$ depend on the linearization around the previous estimate, which in turn depends on the data for all but the first iteration. In the derivation of the posterior covariances, however, we treat the previous estimate as a fixed input around which we linearize, although strictly speaking the previous estimate is itself a random field.

It is convenient to define short-cut notation for the various deviations that occur. In general, we use Y for the true state and $Y^{\eta+1}$ for the estimate. We denote with $\bar{Y}^{\eta+1}$ the prior state as defined in (2.22). For the deviations we define

$$X(t) - \bar{X}^{\eta+1}(t) \equiv X'(t) \quad Y(t) - \bar{Y}^{\eta+1}(t) \equiv Y'(t) \quad (2.30)$$

$$X(t) - X^{\eta+1}(t) \equiv \tilde{X}(t) \quad Y(t) - Y^{\eta+1}(t) \equiv \tilde{Y}(t) \quad (2.31)$$

and from the representer expansion we have

$$X^{\eta+1}(t) - \bar{X}^{\eta+1}(t) = \sum_{k=1}^{N_Z} b_k \Xi^k(t) \quad Y^{\eta+1}(t) - \bar{Y}^{\eta+1}(t) = \sum_{k=1}^{N_Z} b_k \Upsilon^k(t) \quad (2.20)$$

In other words, Y' is the (linearized) prior error and \tilde{Y} is the (posterior) estimation error of the state.

Similarly, the (linearized) true measurement predictions are $M[X^\eta, Y^\eta] + L[X - X^\eta, Y - Y^\eta]$, the prior measurement predictions are $M[X^\eta, Y^\eta] + L[\bar{X}^{\eta+1} - X^\eta, \bar{Y}^{\eta+1} - Y^\eta]$, the estimates for the measured quantities are $M[X^\eta, Y^\eta] + L[X^{\eta+1} - X^\eta, Y^{\eta+1} - Y^\eta]$, and the actual measurements are Z . For the deviations we define

$$v \equiv Z - (M[X^\eta, Y^\eta] + L[X - X^\eta, Y - Y^\eta]) \quad (2.14)$$

$$\hat{v} \equiv Z - (M[X^\eta, Y^\eta] + L[X^{\eta+1} - X^\eta, Y^{\eta+1} - Y^\eta]) \quad (2.32)$$

$$\begin{aligned} \tilde{v} &\equiv (M[X^\eta, Y^\eta] + L[X - X^\eta, Y - Y^\eta]) - (M[X^\eta, Y^\eta] + L[X^{\eta+1} - X^\eta, Y^{\eta+1} - Y^\eta]) \\ &\equiv L[X - X^{\eta+1}, Y - Y^{\eta+1}] \end{aligned} \quad (2.33)$$

Note that $\hat{v} = v + \tilde{v}$. In other words, v is the true measurement error, \tilde{v} is the (posterior) estimation error of the measurement prediction, and \hat{v} is the estimate of the measurement error, or equivalently, the posterior data residual.

Like the representer fields, the deviations defined above are different at each iteration. Again, we opt to drop the superscript $\eta + 1$, as it is clear from the context.

2.4.1 Posterior Covariances

It is a very important fact, not only in the derivation of the posterior error covariances, that the state representer are equal to the linearized (prior) cross-covariances of the measurement predictions and the states.

$$\overline{L_k[X', Y']X'(t)} = \Xi^k(t) \quad \overline{L_k[X', Y']Y'(t)} = \Upsilon^k(t) \quad (2.34)$$

A proof is given in Appendix A.2.1. Using (2.34) it is easy to derive the posterior covariance of the state vector.

$$\begin{aligned}
[C_{\tilde{X}\tilde{X}}(t_1, t_2)]_{ij} &\equiv \overline{\tilde{X}_i(t_1)\tilde{X}_j(t_2)} = [C_{X'X'}(t_1, t_2)]_{ij} - \sum_k \sum_l \Xi_i^k(t_1)[U^{-1}]_{kl}\Xi_j^l(t_2) \\
[C_{\tilde{X}\tilde{Y}}(t_1, t_2)]_{ij} &\equiv \overline{\tilde{X}_i(t_1)\tilde{Y}_j(t_2)} = [C_{X'Y'}(t_1, t_2)]_{ij} - \sum_k \sum_l \Xi_i^k(t_1)[U^{-1}]_{kl}\Upsilon_j^l(t_2) \\
[C_{\tilde{Y}\tilde{X}}(t_1, t_2)]_{ij} &\equiv \overline{\tilde{Y}_i(t_1)\tilde{X}_j(t_2)} = [C_{Y'X'}(t_1, t_2)]_{ij} - \sum_k \sum_l \Upsilon_i^k(t_1)[U^{-1}]_{kl}\Xi_j^l(t_2) \\
[C_{\tilde{Y}\tilde{Y}}(t_1, t_2)]_{ij} &\equiv \overline{\tilde{Y}_i(t_1)\tilde{Y}_j(t_2)} = [C_{Y'Y'}(t_1, t_2)]_{ij} - \sum_k \sum_l \Upsilon_i^k(t_1)[U^{-1}]_{kl}\Upsilon_j^l(t_2)
\end{aligned} \tag{2.35}$$

The prior error covariance is denoted with $C_{X'X'}(t_1, t_2)$ etc. Details of the derivation are presented in Appendix A.2.2. The estimation error covariance for the state indicates how close to the truth the estimator thinks it is. Its diagonal provides us with error bars around the state estimate.

The posterior covariance for the estimates of the measured variable at observation times is

$$C_{\tilde{v}} \equiv \overline{\tilde{v}\tilde{v}^T} = R - RU^{-1}R \tag{2.36}$$

Details of the derivation can be found in Appendix A.2.2. Again, $C_{\tilde{v}}$ indicates how close to the truth the estimator thinks it is. We can use the diagonal of $C_{\tilde{v}}$ to draw error bars around the estimates of the measurements.

Finally, we find for the covariance of the posterior data residuals

$$C_{\hat{v}} \equiv \overline{\hat{v}\hat{v}^T} = C_{\tilde{v}} + C_v - C_v U^{-1}R - (C_v U^{-1}R)^T \tag{2.37}$$

For details of the derivation see Appendix A.2.2. We can use $C_{\hat{v}}$ to normalize the data residuals after the estimation and to check the algorithm's underlying statistical assumptions. This test is particularly powerful when field data are assimilated, because it provides a way to check whether the estimator is operating in accordance with its statistical assumptions. It must be stressed, however, that this test relies on the linearizations that have been applied. Since the model is nonlinear, both \tilde{v} and \hat{v} already have bias (e.g. $\overline{M[X, Y]} \neq M[\overline{X}, \overline{Y}]$).

Even if we are only interested in the covariance of the posterior data residuals (2.37), we need a good approximation of the matrix $U^{-1} = (R + C_v)^{-1}$ (2.27). This task is very computationally demanding, because we need to calculate individual representer functions. Luckily, we only have to do this once. After the iteration on the nonlinearity of the forward model has converged, we can switch from the indirect representer technique to computing individual representers as needed. Depending on the problem at hand, we will likely be able to get a reasonable approximation of U^{-1} by calculating only a subset of all representer functions. Note that data compression (Section 2.5) could be particularly helpful to reduce the number of individual representers and therefore the dimension of U .

2.4.2 Prior Covariance

In order to get error bars around the state estimate or to reinitialize the algorithm for subsequent assimilation windows, we need to compute at least the posterior variance of the

state, which is given by the diagonal elements of the estimation error covariance (2.35). Although the storage of the N_Z representer fields $\Upsilon^k(t)$ poses a serious problem, a difficult part in computing the estimation error variance is the *prior* variance of the state, namely $[C_{X'X'}(t, t)]_{ii} \equiv \sigma_{X'_i}^2(t)$ and $[C_{Y'Y'}(t, t)]_{ii} \equiv \sigma_{Y'_i}^2(t)$.

For the initialization of a subsequent assimilation interval, we are only interested in the posterior error covariance of the state at the final time of the previous assimilation window. This also means that we only need the prior error covariance at the final time. If we only need the prior state covariance at certain times and for certain states, we can again use the representer technique.

Introducing a pseudo-observation for a direct measurement of the state $Y_j(t_f)$ at the final time allows us to compute the corresponding pseudo state representer function Υ^{*j} . Now recall that the state representer function is the covariance of the state with the measurement prediction, which in the case of our pseudo-observation is the state at the final time (Sections 2.4.1 and A.2.1). Therefore, the pseudo representer function $\Upsilon^{*j}(t_f)$ at the final time includes the prior state cross-covariance at the final time.

$$[C_{Y'Y'}(t_f, t_f)]_{ij} = \Upsilon_i^{*j}(t_f) \quad (2.38)$$

It is important to note that the computation of the pseudo state representer function Υ^{*j} does *not* require the specification of a corresponding measurement error covariance or a pseudo measurement value. In the representer technique, all the prior knowledge is encapsulated in the representer functions and the representer matrix, which are completely separate from the measurement error covariance. In fact, by comparing (2.9) and (2.24) we see that the representer functions are computed using unit impulses instead of the data misfit terms in the backward equation.

2.5 Data Compression and Observing System Assessment

Satellite images often contain a lot of redundant information. When such images are transmitted, image compression is routinely used to minimize the number of bits being transferred. Similarly, treating each pixel of the image as a separate data point in the assimilation algorithm may be very inefficient.

Whether or not data compression techniques help reduce the computational effort required by the data assimilation scheme depends on the assimilation method. If the objective function (2.5) is minimized directly with a gradient search technique, the search happens effectively in the space of the uncertain inputs. If model error is included, this space is more or less equivalent to the state space and hence very big (Section 8.3). Moreover, compressing the data does not lead to significant computational savings because the state space remains unchanged when the data are compressed.

In the representer technique, on the other hand, the search consists mainly of solving (2.26) for the representer coefficients, where the dimension of the space involved is equal to the number of observations. We can say that the search is carried out in the data space and that the unobservable modes have been discarded a priori [Bennett, 1992].

Since in land-surface data assimilation there are generally a lot fewer observations than states, the representer method is an elegant way to reduce the size of the estimation problem by searching only the data space. In addition, the representer method offers the intriguing possibility to save even more computational effort by compressing the data a priori [Bennett

et al., 1993]. If each satellite image is compressed linearly, for example with a singular value decomposition or a two-dimensional Fourier Transform, the assimilated data vector Z can simply be defined as a collection of linear combinations of the original observations.

The degree of meaningful compression and the corresponding savings depend of course on the problem at hand. Note that data compression does not generally mean that observations are discarded. In fact, we typically use all the original observations, but instead of looking at the individual measurements which are supplied by the sensor, we focus on the dominant linear combinations.

It is also important to emphasize that the indirect representer method and a priori data compression may achieve related savings. Data compression might therefore not be of much additional help if the estimates are computed with the indirect representer method (Section 2.3.4). In contrast, only data compression may make the calculation of the posterior error covariances feasible, because for the posterior covariances the individual representer functions are needed (Section 2.4.1), and their number is equal to the length of the (compressed) data vector Z . Results related to this question are presented in Section 8.2.1.

Within the (direct) representer technique, we can also assess the observing system a posteriori by carrying out an eigenvalue decomposition of the matrix $U \equiv (C_v + R)$ (2.27). Note that the inverse is unstable if the matrix U is poorly conditioned. If this is the case, the measurement functional could be rotated and the modes corresponding to very small eigenvalues could be discarded to stabilize the assimilation algorithm. This technique amounts to compressing the data a posteriori. For details see [Bennett, 1992].